1. Shares

Standard Bank

Introduction

You have written a computer program that you are confident will be able to accurately predict the stock market. You want to put this knowledge to work, but as you are cautious, you will limit yourself to purchasing shares from a single company.

Task

You will be given the prices of N shares, at two different times. You will buy the largest whole number of shares that you can afford from a single company at the first price and then sell them at the second price. You should report how much money you can make if you do this optimally.

Example

Suppose that there are shares from four different companies: A, B, C and D. Initially, these have share prices of R10, R20, R15 and R51 respectively. Your program predicts that later the prices will be R12, R23, R14 and R70. If you have R100, you can buy 10 shares from company A at R10 and then sell them at R12 to make R20 profit, and this is the largest profit you can make.

Input

The first line of input contains an integer T, the number of test cases. The first line of each test case contains two space-separated integers: N and C, respectively the number of different stocks and the amount of money you have in rands. The next two lines each contain N space-separated positive integers, the share prices of each stock in rands at the first and second time respectively.

Sample input

```
2
4 100
10 20 15 51
12 23 14 70
4 103
10 20 15 51
12 23 14 70
```

Output

For each test case, output a line of the form

```
Case #X: P
```

where X is the test case number, starting from 1, and P is the best possible profit in rands.

Sample output

```
Case #1: 20
Case #2: 38
```

Constraints

In the input used to test your program, the following constraints will hold:

- $1 \leq T \leq 20$
- $1 \leq N \leq 10\,000$
- $0 \leq C \leq 10\,000$
- $1 \leq$ each share price $\leq 10\,000$

Time limit

5 seconds

Memory limit

128 MiB

2. Copying Books

Standard Bank

Introduction

Johannes Gensfleisch zur Laden Gutenberg was a German goldsmith who achieved fame for his invention of movable-type printing. As his invention became famous, orders for the printing of books poured in. With too much work to handle he hired Fusstuch zur Linuburg to help him.

Fusstuch's work consisted of composing cast metal letter types into lines of text. Fusstuch became lazy and glued letter types together to form word types. This accelerated the process since it eliminated the need for recomposing the same words over and over again. Fusstuch was very pleased with his increased productivity, but after a while he did not have any individual letter types left to form new words. Help Fusstuch to determine how many lines of the book can be printed using his set of precomposed word types when he prints the book line-by-line.

Task

Given a set of word types and the content of the book, determine how many lines of the book can be composed from the set of word types. Each word type can be used only once per line. A word may not be formed by combining multiple word types.

Example

Suppose two word types are available for the word `lorem`, and one word type is available for each of the words `dolor`, `ipsum`, `lorem` and `sit`. Suppose the book contains four lines of text:

```
lorem ipsum dolor lorem
sit amet
sit sit
loremipsum
```

The first line can be printed because a word type is available for every word, but the second line cannot be printed since there is no word type available for the word `amet`.

There is only one word type available for the word `sit`, but the word is used twice within the third line. For this reason, the third line cannot be printed.

The fourth line cannot be printed since there is no type available for the word `loremipsum`, and a word may not be formed by concatenating types.

Input

The first line of input contains the integer T, the number of test cases. This is followed by T test cases.

The first line of each test case contains two space-separated integers: W and L, representing the number of word types and the number of lines in the book respectively.

The next W lines will each contain a single word made up of lower case letters, describing the set of word types. This is followed by L lines describing the content of the book, each of which is a space-separated list of lower case words. None of the lines in the input will contain leading, trailing or repeated spaces.

Sample input

```
1
5 4
dolor
ipsum
lorem
lorem
sit
lorem ipsum dolor lorem
sit amet
sit sit
loremipsum
```

Output

For each test case, output one line of the form

```
Case #X: N
```

where X is the test case number, starting from 1 and N is the number of lines that can be printed with the use of the word types.

Sample output

```
Case #1: 1
```

Constraints

In the input used to test your program, the following constraints will hold:

- $1 \leq T \leq 10$
- $1 \leq W \leq 5\,000$
- $1 \leq L \leq 2\,000$
- $1 \leq$ number of words in a book $\leq 100\,000$
- $1 \leq$ length of any word in the input ≤ 10

2. Copying Books

Time limit

2 seconds

Memory limit

128 MiB

3. Robot Testing

Introduction

Amy and Bongani are busy testing the robot that they are building together. The intention is for the robot to one day take over the world, but, being a work in progress, the robot only has limited functionality thus far. They have devised a game that allows them to take turns controlling the robot, whilst testing all of its functionality. The game works as follows:

Bongani places the robot on a chosen floor tile in their workshop, and then allows Amy to decide whether she would like to send the first instruction or the second. Thereafter they take turns sending instructions to the robot. The available instructions are to make the robot move:

- one tile south
- one tile west
- one tile south-west

The player who issues the last instruction, moving the robot into the south-west corner, loses the game, and is required to retrieve the robot.

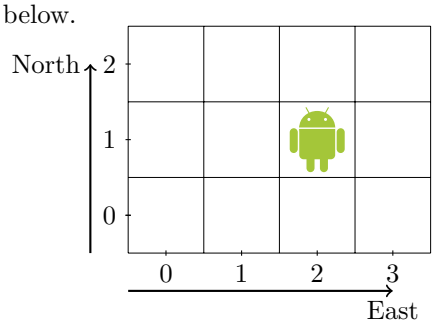
Amy realises that by playing optimally, she can guarantee her victory as long as she correctly chooses to go first or second. She asks you to write a program that will be able to tell her whether to go first or second. Once she knows that, she is able to figure out which moves to make on her own.

Task

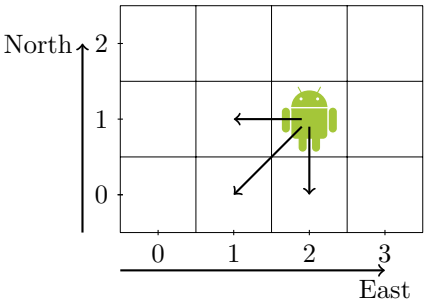
Given the starting tile of the robot, calculate whether the first or second player would have a winning strategy.

Example

Suppose Bongani placed the robot at a position 1 tile north of, and 2 tiles east of the corner tile as indicated below.



From that position, the first player has three possible moves, as indicated in the diagram.



If the first player moves the robot diagonally south-west to the tile one tile east of the corner tile, then the second player would have only one move remaining, to move the robot one tile west into the corner tile. Hence Bongani's placement of the robot has a winning strategy for the first player, and Amy should choose to go first.

Input

The first line of input contains the integer T, the number of test cases. This is followed by T lines representing each test case.

Each test case contains one line with two space separated integers: N and E, representing the number of tiles north of –, and the number of tiles east of the corner block, respectively.

An input where $N = 0$ and $E = 0$ would represent the corner block itself, and is therefore not a valid starting position.

Sample input

```
3
1 2
0 1
2 2
```

Output

For each test case, output one line of the form

```
Case #X: R
```

where X is the test case number, starting from 1 and R is the text "FIRST" if the starting player has a winning strategy, or the text "SECOND" if the second player has a winning strategy.

Sample output

```
Case #1: FIRST
Case #2: SECOND
Case #3: SECOND
```

3. Robot Testing

Constraints

In the input used to test your program, the following constraints will hold:

- $1 \leq T \leq 20$
- $0 \leq N \leq 40\,000$
- $0 \leq E \leq 40\,000$
- $1 \leq N + E$

Time limit

5 seconds

Memory limit

128 MiB

4. Collecting Stones

Standard Bank

Introduction

A geologist has asked you to help optimise the collection of loose stones from a narrow canyon. As she moves through the canyon, she sees stones of various weights, which she may distribute amongst two buckets that she holds in either hand.

She wishes to collect the largest total mass of stones she can. The total weight of the stones is not too much for her to carry. However, if one bucket is much heavier than the other, she is thrown off balance and cannot progress. Additionally, since she wishes to move through the canyon quickly, she does not return for any stones she leaves behind, nor does she transfer stones from one bucket to the other after she has collected them.

Task

You are given the weights of the stones in the order they are encountered and the maximum weight difference between the two buckets that the geologist can handle. You should report the total weight of stones the geologist could collect if she collects the stones in the best possible way.

Example

Suppose the maximum weight difference is 4 kg and the stones are of weights 3 kg, 4 kg, 8 kg, 1 kg, 2 kg and 4 kg. An optimal strategy would be to ignore the 3-kg stone, place the 4-kg stone in the left bucket and the 8-kg stone in the right bucket, before proceeding to place the rest of the stones in the left bucket. The total weight of collected stones is thus $4 + 8 + 1 + 2 + 4 = 19$ kg.

Note that if the geologist were to instead pick up both of the first two stones, she would need to place them in separate buckets and would then be unable to collect the 8-kg stone without becoming unbalanced.

Input

The first line of input contains an integer T, the number of test cases. The first line of each test case contains two space-separated integers: N and K, the number of stones in the canyon and the maximum allowed difference between the weight in each bucket, respectively. The next line contains N space-separated positive integers, the weights of the stones in the order they will be encountered.

Sample input

```
1
6 4
3 4 8 1 2 4
```

Output

For each test case, output a line of the form

```
Case #X: M
```

where X is the test case number, starting from 1, and M is the largest total weight of stones that could be collected.

Sample output

```
Case #1: 19
```

Constraints

In the input used to test your program, the following constraints will hold:

- $1 \leq T \leq 13$
- $1 \leq N \leq 10\,000$
- $0 \leq K \leq 150$
- $1 \leq$ the weight of each stone $\leq 2\,000$

Time limit

6 seconds

Memory limit

128 MiB

5. Missing aeroplane

Standard Bank

Introduction

An aeroplane has gone missing, and you have been approached by the international search team to help locate it. All the aeroplane's communication systems were disabled, except for a device that transmits pulse signals to ground stations. Fortunately, some of the radio signals transmitted by the missing aeroplane were recorded.

The only information available are timestamps identifying when each of the signals arrived at each of the ground stations. The signals arrived at different times due to the aeroplane being a different distance from each of the ground stations. The time at which the pulses were emitted is not known.

Task

Given the location of four ground stations, and the arrival time of the signals at each of the four ground stations, determine for each signal the position of the aeroplane when it transmitted that signal.

Assume a flat world where the variation in elevation may be ignored. You may assume that the signals were radiated equally in all directions, and that they propagated at a speed of $c = 0.3$ metre per nanosecond (m/ns). The time it takes the signal to propagate a distance d metres is $t = \frac{d}{c}$ ns.

You may assume that the time of arrival recorded by the ground stations and the distribution of the ground stations are such that a unique location can be determined for each signal sent by the aeroplane.

Example

Suppose ground stations b_1, b_2, b_3 and b_4 are located at $(0, 0)$, $(114, -72)$, $(60, 120)$ and $(36, 18)$ respectively, where x and y in (x, y) denotes a coordinate in metres.

Suppose a signal transmitted from the aeroplane was received by b_1 at time $t_1 = 250$ ns, by b_2 at time $t_2 = 350$ ns, by b_3 at time $t_3 = 450$ ns, and by b_4 at time $t_4 = 150$ ns.

The only possible location of the aeroplane is $(60, 0)$, with the signal being transmitted at time $t = 50$ ns.

Input

The first line of input contains the integer T, the number of test cases. This is followed by T test cases.

The coordinates of the ground stations are specified within the first four lines of each test case. Each line contains two integers, x_i and y_i, denoting the position of a ground station.

The next line contains a single integer N, the number of signals transmitted by the aeroplane. Each of the next N lines contains four floating-point numbers, the arrival time of signal s_i at each of the four ground stations, in the order the ground stations' coordinates were specified.

All coordinates are in metres, and all timestamps are in nanoseconds (1×10^{-9} s).

Sample input

```
1
0 0
114 -72
60 120
36 18
2
250.0 350.0 450.0 150.0
203.548 607.204 454.103 221.335
```

Output

For each test case, output a line of the form

```
Case #C:
```

where C is the test case number (starting from 1). Then output N lines, one for each signal s_i, in the order they appeared in the input. Each line must be in the form

```
X Y
```

where X and Y are integers denoting the location of the aeroplane when it transmitted the signal, given to the nearest metre.

Sample output

```
Case #1:
60 0
2 31
```

Constraints

In the input used to test your program, the following constraints will hold:

- $1 \leq T \leq 10$
- $1 \leq N \leq 10\,000$
- $-1\,000\,000 \leq x, y \leq 1\,000\,000$
- $0 \leq$ time of transmission and reception $\leq 10\,000\,000$
- A signal will not arrive at the same time at more than one station.
- The output will not change when each timestamp in the input is changed by up to 10^{-6}.

5. Missing aeroplane

Time limit

5 seconds

Memory limit

128 MiB